Noname manuscript No.
(will be inserted by the editor)

Multistart with early termination of descents

Antanas Žilinskas, Jonathan Gillard, Megan Scammell, Anatoly Zhigljavsky

Abstract Multistart is a celebrated global optimization technique frequently applied in practice. In its pure form, multistart has low efficiency. However, the simplicity of multistart and multitude of possibilities of its generalization make it very attractive especially in high-dimensional problems where e.g. Lipschitzian and Bayesian algorithms are not applicable. We propose a version of multistart where most of the local descents are terminated very early; we will call it METOD as an abbreviation for Multistart with Early Termination Of Descents. The performance of the proposed algorithm is demonstrated on randomly generated test functions with 100 variables and a modest number of local minimizers.

Keywords global optimization; random search; multistart; statistical inference

1 Introduction

Multistart is, seemingly, the oldest global optimization method. Although frequently claimed inefficient, it is nevertheless often used in various applications (see for example [4], [5] and [9]). The main advantage of multistart is its simplicity and clear interpretability of results. Multistart remains attractive for researchers in various applied areas where optimization problems are multi-extremal and high-dimensional. The known theoretically well-substantiated methods (e.g. Lipschitzian and Bayesian) which are efficient for the problems of small or sometimes modest dimensionality are not appropriate for high-dimensional problems in view of inherent complexity of the corresponding algorithms. The choice between multistart and a heuristic algorithm depends on many details of the available information about the optimization problem in question.

Repeated descents to the same local minimizers is the obvious disadvantage of multistart. There were numerous extensions of the method which reduce this disadvantage. A heuristic termination condition is the stagnation of search caused by the repeated descents to the found minimizers. The redundant descents could be avoided if the fact of finding of the global minimum could be indicated with a reasonable accuracy.

A. Žilinskas
Institute of Data Science and Digital Technologies,
Vilnius University, Akademijos 4, Vilnius, LT 08663, Lithuania
E-mail: antanas.zilinskas@mii.vu.lt
J. Gillard, M. Scammell, A. Zhigljavsky
School of Mathematics, Cardiff University, Cardiff CF24 4AG, UK
E-mail: GillardJW@cardiff.ac.uk, ScammellM@cardiff.ac.uk, ZhigljavskyAA@cardiff.ac.uk

The other direction to enhance the efficiency of multistart is the termination of local descents which approach already found minimizers before they are stopped by conventional conditions based on local optimality. The clustering aided implementation of this idea was started by A.Törn; for the early investigations see [10], and for the later results and references see [11]. Recent discussions on the use of multistart are included in [3] and [6]. In this paper, we further develop this idea and demonstrate that some local descents can be stopped very early, well before they get close to a local minimizer.

Any optimization algorithm cannot be most efficient in all possible cases. We aim at the global optimization problems with a black-box objective function $f(\cdot)$ characterized by the following properties:

(a) the feasible domain $\mathfrak{X}$ is high-dimensional but has simple structure; in numerical studies we will assume $\mathfrak{X} = [0,1]^d$ with $d = 100$;
(b) $\|\nabla f(x)\| \neq 0$ for almost all $x \in \mathfrak{X}$, where $\nabla f(x)$ is the gradient of $f(\cdot)$;
(c) computation of the objective function values and its derivatives is not expensive;
(d) the total number of local minimizers is not very large;
(e) the volume of the region of attraction of the global minimizer is not very small.

The paper is organized as follows. In Section 2 we prove the key result showing that for a quadratic function and two arbitrary points the gradients force the two points to move closer to each other. This will be our main base for making early terminations of local descents. In Section 3 we formulate our main algorithm called METOD (Multistart with Early Termination Of Descents), discuss its properties and modifications and suggest a reasonable choice of the algorithm's parameters. In Section 4 we describe several statistical procedures which can be used for devised intelligent stopping rules in METOD. Finally, in Section 5 we provide results of numerical studies on 100-dimensional test functions.

2 Monotonicity of descent trajectories

2.1 The main result

Assume we have a quadratic function

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c\,, \quad x \in \mathbb{R}^d, \tag{1}$$

where A is a positive definite $d \times d$ matrix, b is some vector in $\mathbb{R}^d$ and c is some constant.

The gradient of $f(\cdot)$ at x is $\nabla f(x) = Ax + b$. In this case, given a point $x_k \in \mathbb{R}^d$, a k-th iteration of a gradient descent algorithm would return the point

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k) = x_k - \gamma_k(Ax_k + b)\,, \tag{2}$$

where $\gamma_k \geq 0$ is some step-size.

Theorem 1. *Assume that f is a quadratic function* (1)*, where A is a positive definite matrix. Let x and y be two arbitrary points in $\mathbb{R}^d$ such that $\|\nabla f(x)\| > 0$ and $\|\nabla f(y)\| > 0$. Fix some $\beta > 0$ and define*

$$\begin{cases} \tilde{x} = x - \beta \nabla f(x)\,, \\ \tilde{y} = y - \beta \nabla f(y)\,; \end{cases} \tag{3}$$

that is, we apply the rule (2) *to the points* x *and* y *with the same step-size* β*. If* $\beta < 1/\lambda_{\max}$*, where* $\lambda_{\max}$ *is the maximal eigenvalue of the matrix* A*, then*

$$\|\tilde{x} - \tilde{y}\| < \|x - y\|\,, \tag{4}$$

where $\|\cdot\|$ *denotes the Euclidean norm of a vector in* $\mathbb{R}^d$*.*

Proof. We have

$$\tilde{x} - \tilde{y} = [x - \beta(Ax + b)] - [y - \beta(Ay + b)] = (I_d - \beta A)(x - y) \tag{5}$$

where I_d is the identity matrix of size $d \times d$. Since $0 < \beta < 1/\lambda_{\max}$, the matrix $I_d - \beta A$ is positive definite and $I_d - \beta A < I_d$, where the inequality $<$ means that the difference between the matrices in the right-hand and left-hand sides is positive definite. As $I_d - \beta A < I_d$ we also have $(I_d - \beta A)^2 < I_d$.

For arbitrary x and y in $\mathbb{R}^d$ with $\|\nabla f(x)\| > 0$ and $\|\nabla f(y)\| > 0$ we obtain from (5):

$$\|\tilde{x} - \tilde{y}\|^2 = (\tilde{x} - \tilde{y})^T(\tilde{x} - \tilde{y}) = (x - y)^T(I_d - \beta A)^2(x - y) < (x - y)^T(x - y) = \|x - y\|^2\,,$$

where the inequality above is a consequence of $(I_d - \beta A)^2 < I_d$. □

We will call the points $\tilde{x}$ and $\tilde{y}$ computed by the rule (3) 'the partner points associated with x and y respectively'. Theorem 1 can be interpreted as saying that if the objective function f is quadratic and the coefficient β in (3) is small enough then for arbitrary x and y the associated partner points $\tilde{x}$ and $\tilde{y}$ are always closer to each other than the original points x and y.

2.2 Using gradients of different functions

Let us now discuss what happens when the partner points $\tilde{x}$ and $\tilde{y}$ are computed for gradients of two different functions.

Assume we have two quadratic functions

$$f_i(x) = \frac{1}{2}x^T A_i x + b_i^T x + c_i \;\; (i = 1, 2)\,, \;\; x \in \mathbb{R}^d,$$

where A_1 and A_2 are two different non-negative definite $d \times d$ matrices, b_1 and b_2 are two vectors in $\mathbb{R}^d$ and c_1, c_2 are some constants.

For two arbitrary points x and y in $\mathbb{R}^d$ define their partner points by

$$\begin{aligned} \tilde{x} &= x - \beta\nabla f_1(x) = x - \beta(A_1 x + b_1)\,, \\ \tilde{y} &= y - \beta\nabla f_2(y) = y - \beta(A_2 y + b_2)\,. \end{aligned}$$

Then

$$\tilde{x} - \tilde{y} = (x - y) - \beta(A_1 x + b_1 - A_2 y - b_2)\,.$$

If we impose some natural randomness assumptions on either points x and y, vectors b_1 and b_2 or matrices A_1 and A_2 then we may observe that the inequality (4) holds with probability much smaller than 1.

2.3 Generalization of Theorem 1 to the case when the derivatives are computed in a few directions only

Since our main range of optimization problems is high-dimensional, computing full gradients for performing local descents can be costly and in modern literature [8] it is suggested at a given iteration to compute derivatives only in very few directions (different at different iterations). A generalization of Theorem 1 to this case is as follows.

Theorem 2. *Assume that f is a quadratic function* (1)*, where A is a non-negative definite matrix and $0 \le \beta \le 1/\lambda_{\max}$. Let x and y be two arbitrary points in $\mathbb{R}^d$ and*

$$\begin{cases} \tilde{x} = x - \beta[\nabla f(x)]_{i_1,\dots,i_k} \\ \tilde{y} = y - \beta[\nabla f(y)]_{i_1,\dots,i_k}\,, \end{cases}$$

where for any $1 \le i_1 < \ldots < i_k \le d$ and a vector $a = (a_1, \dots, a_d)^T \in \mathbb{R}^d$ we define the vector $a_{i_1,\dots,i_k} \in \mathbb{R}^d$ as a vector $u = (u_1, \dots, u_d)^T$ with components

$$u_i = \begin{cases} a_i, \text{ if } \ i \in \{i_1, \dots, i_k\} \\ 0, \text{ otherwise.} \end{cases}$$

Then

$$\|\tilde{x} - \tilde{y}\| \le \|x - y\|\,. \tag{6}$$

Proof. Similar to (5) we have

$$\tilde{x} - \tilde{y} = (x - y) - \beta[A(x - y)]_{i_1,\dots,i_k} = (I_d - \beta A_{i_1,\dots,i_k})(x - y)$$

where $A_{i_1,\dots,i_k} = (\tilde{a}_{i,j})_{i,j=1}^d$ is a $d \times d$ matrix with elements $\tilde{a}_{i,j}$ defined as follows:

$$\tilde{a}_{i,j} = \begin{cases} a_{i,j}, \text{ if } \ i, j \in \{i_1, \dots, i_k\} \\ 0, \ \text{ otherwise.} \end{cases}$$

Since the matrix $I_d - \beta A_{i_1,\dots,i_k}$ is non-negative definite, the proof follows. □

Remarks.

(a) The inequality (4) is strict but the inequality (6) is not.
(b) Depending on the set $I = \{i_1, \dots, i_k\}$, the probability that the inequality (6) is true when derivatives are computed for two different functions as in Section 2.2, can be significantly larger than $\frac{1}{2}$.
(c) It is important that the set of indices $I = \{i_1, \dots, i_k\}$, where the derivatives of f at both points are computed, is the same. If the sets are different, the inequality (6) may not hold. This is the case even if one set of indices is a subset of the other one: assume, as an example, $d = 2$, the function $f(x) = x^T Ax$ with $A > 0$ and two points: $x \in \mathbb{R}^2$ and $y = cx$ with $c > 1$. Using any non-empty set of indices for x and the empty set for y (so that $\tilde{y} = y$) we clearly have $\|\tilde{x} - \tilde{y}\| > \|x - y\|$.
(d) Assume that we compute derivatives for x and y for the sets of indices $I = \{i_1, \dots, i_k\}$ and $J = \{j_1, \dots, j_l\}$ respectively. Then we can apply Theorem 2 and therefore the inequality (6) for the set of indices $I \cap J$, the intersection of I and J.

3 Description of the algorithm

3.1 Notations

- β and δ: small positive constants.
- N: number of starting points;
- $x_n = x_n^{(0)} \in \mathfrak{X}$: random starting points uniformly distributed on $\mathfrak{X}$ $(n = 1, 2, \ldots, N)$;
- K_l: the number of local decent iterations needed to get close to l-th local minimizer;
- $x_n^{(k)}$: the n-th point after k iterations;
- steepest descent iteration:

$$x_n^{(k+1)} = x_n^{(k)} - \gamma_n^{(k)} \nabla f(x_n^{(k)}) \tag{7}$$

  where $\gamma_n^{(k)} = \text{argmin}_{\gamma>0} f(x_n^{(k)} - \gamma \nabla f(x_n^{(k)}))$; if the function $h(\gamma) = f(x_n^{(k)} - \gamma \nabla f(x_n^{(k)}))$ has several minimizers in the region $\gamma \in (0, +\infty)$, then the local minimizer with smallest value of γ is used;
- partner point associated with $x_n^{(k)}$:

$$\tilde{x}_n^{(k)} = x_n^{(k)} - \beta \nabla f(x_n^{(k)})\,; \tag{8}$$

- M: the minimum number of the steepest descent iterations applied starting at each initial point $x_n = x_n^{(0)}$ $(n = 1, 2, \ldots, N)$;
- l: index for the local minimizers and regions of attraction, $l = 1, ..., L$; here L is the total number of regions of local minimizers found so far;
- x_l^*: l-th local minimizer $(l = 1, ..., L)$;
- A_l: notation for the l-th region of attraction $(l = 1, ..., L)$;

3.2 METOD: Multistart with Early Termination Of Descents

1. **Initialization.**
   Generate a uniformly distributed point $x_1 = x_1^{(0)} \in \mathfrak{X}$. Use iterations (7) to find a local minimizer x_1^*. Stop iterations at the smallest $k = K_1$ such that $\|\nabla f(x_1^{(k)})\| < \delta$.
   For all points $x_1^{(k)}$ computed in (7) with $k = M-1, M, \ldots, K_1$ compute the associated partner points using (8).
   Set $n := 2$, $L := 1$.
2. **n-th iteration.**
   1. *Initial point.* Generate a uniformly distributed point $x_n = x_n^{(0)} \in \mathfrak{X}$.
   2. *Warming-up period.* Compute $x_n^{(j)}$ $(j = 1, \ldots, M)$ by applying M iterations (7) starting at $x_n^{(0)}$.
   3. *Checking if x_n belongs to one of previously identified regions of attractions.* Using (8) compute $\tilde{x}_n^{(M-1)}$ and $\tilde{x}_n^{(M)}$, the partner points associated with $x_n^{(M-1)}$ and $x_n^{(M)}$ respectively. For all $l = 1, \ldots, L$ check the conditions

$$\|\tilde{x}_n^{(M)} - \tilde{x}_l^{(i)}\| < \|x_n^{(M)} - x_l^{(i)}\| \;\text{ and }\; \|\tilde{x}_n^{(M-1)} - \tilde{x}_l^{(i)}\| < \|x_n^{(M-1)} - x_l^{(i)}\| \tag{9}$$

   for all $l = 1, \ldots, L$ and $i = M-1, M, \ldots, K_l$.
   If for a given l the inequalities (9) hold for all $i = M-1, M, \ldots, K_l$ then we presume that x_n may belong to the region of attraction A_l.
   Let S_n be the set of indices l such that x_n may presumably belong to A_l.

4. *Making the decision of whether to continue the descent.*
   (4a) If the set S_n contains exactly one index l then we terminate iterations (7) which have started at x_n and assign x_n to A_l.
   (4b) If the set S_n contains more than one index l then we assign x_n to the attraction region A_l which corresponds to the local minimizer x_l^* closest to the point $x_n^{(M)}$.
   (4c) If the set S_n is empty then we continue iterations (7) which have started at x_n. Find a local minimizer x_{l+1}^*. Stop iterations at the smallest $k = K_{l+1}$ such that $\|\nabla f(x_n^{(k)})\| < \delta$. For all points $x_n^{(k)}$ computed in (7) with $k = M-1, M, \ldots, K_{l+1}$ compute the associated partner points using (8). Set $L := L + 1$.

3. **Stopping rule.** Check a stopping rule (for example, $n = N$). If there is no decision made to stop the algorithm, then set $n := n + 1$ and return to the main iteration 2.

3.3 Comments and choice of parameters

- Choice of β should be made as discussed in Theorem 1. If some regions of attractions are badly defined (very long in some directions and short in other ones) then β should be very small. Generally, $\beta = 0.01$ would work well in typical problems.
- δ is needed to terminate steepest descent iterations. We recommend $\delta = 0.001$ or similar. If one requires higher precision for determining local minimizers then this can either be done after termination of the main algorithm or in the process of running the algorithm; we simply suggest not to take into account the iterations of the steepest descent when the norm of the gradient becomes too small as this can cause numerical instability related to conditions of Theorem 1. A detailed study of the convergence of various gradient algorithms is available in [1].
- The algorithm may have difficulties when some of local minimizers are very easy to compute (that is, when some regions of attractions are too round and the steepest descent finds the corresponding local minima very fast). In this case, the values K_l will be very small and the checking conditions (9) will be unreliable. One of a number of natural suggestions to combat this would be to avoid checking (9) for l with small values of K_l; in this case, we may descend to this local minima many times but it would not be too costly as these descents are fast and hence cheap.
- For any starting point x_n, it may happen that $\|\nabla f(x_n)\| < \delta$. This may happen either if x_n is very close to a local minimizer (in this case we do not perform any iterations) or local maximizer (in this case, make any first iteration which takes you out of the small vicinity of the maximizer).
- We have introduced the parameter M for making a warming-up period at the start of all descents. This is done for the following reason: if an initial point x_n is located very far from a local minimizer and somewhere in-between two different regions of attractions then the objective function may be locally non-convex and certainly very far from being locally quadratic. The first few steepest descent iterations are often very long and very efficient (the convergence of the steepest descent slows down as iterations progress). By introducing the warming-up period we give iterations time to arrive at a distant vicinity of the related local minimizer where the conditions of Theorem 1 start to be approximately applicable. Our recommendation is to choose $M = 3$ or $M = 4$. This seems to be sufficient in all our practical experiments. Reducing M down to 1 may cause too many decisions based on (9) made in the conditions when the region with local quadraticity is far from being reached. On the other hand, choosing larger values of M would decrease the computational efficiency of the algorithm.

- In (9), we require checking the monotonicity conditions of Theorem 1 for two consecutive points, $x_n^{(M-1)}$ and $x_n^{(M)}$. This is related to the fact that steepest descent iterations often approach local minimizers in a zig-zag manner, see [7, Chapter 7].
- Of course it may happen that we descend to the same local minimizer several times without making an early stop of the iterations. This may happen if M is either too small (local quadraticity has not been achieved and the monotonicity condition of Theorem 1 is not seen) or too large (leading to decrease of values K_l and hence to smaller number of necessary conditions (9) to be checked). Descending to the same local minimizers without early stops makes the algorithm less computationally efficient but does not lead to any decision errors as long as at the end of computations we check all found local minimizers and join the ones which are very close.
- A more serious issue is the mistake when we assign an initial point to a wrong region of attraction. This mistake can easily happen if some of the values K_l are too small (see one of the first remarks above) or simply when some regions are non-convex even in the small neighbourhood of local minimizers. One of the advices on how to reduce errors of this type would be to add an extra condition of closeness of $x_n^{(k)}$ to x_l^* when assigning an initial point x_n to A_l. Of course, if this closeness condition would be too strong then it would dominate the main conditions (9) and the algorithm would become very similar to a standard multistart algorithm with clustering.
  We shall account for this kind of error in decision making when we shall talk about statistical procedures in Section 4.
- In view of Theorem 2 the algorithm METOD can be generalized to the case when at each iteration of a local descent the derivatives of f are computed only in one or several directions.
- In the algorithm METOD above we used the steepest descent iterations (7) for local descent. Any other local descent algorithm can also be used; the only essential part of the suggested methodology is the use of the partner points (8).

4 Statistical inferences in random multistart and algorithm METOD

In this section, we recall some results of R. Zieliński published in a seminal paper [13], which can be used to devise intelligent stopping rules in METOD which is a variation of a random multistart; that is of a multistart method with i.i.d. random initial points. Note that there has been very little progress in the area of making statistical inferences in random multistart since 1981, the time of the publication of [13]; see a short literature review in [12, Section 2.6.2].

Assume that $\mathrm{vol}(\mathfrak{X})=1$, $f(\cdot)$ has a finite but unknown number l of local minimizers $x_*^{(1)}, \ldots, x_*^{(l)}$, and $\mathcal{D}$ be a local descent algorithm. We write $\mathcal{A}(x) = x_*^{(i)}$ for $x \in \mathfrak{X}$, if starting at the initial point x the algorithm $\mathcal{D}$ leads to the local minimizer $x_*^{(i)}$. Set $\theta_i = \mathrm{vol}(A_i)$ for $i = 1, \ldots, l$, where $A_i = \{x \in \mathfrak{X} : \mathcal{D}(x) = x_*^{(i)}\}$ is the region of attraction of $x_*^{(i)}$ It follows from the definition that $\theta_i > 0$ for $i = 1, \ldots, l$ and $\sum_{i=1}^{l} \theta_i = 1$.

The simplest version of random multistart is the following primitive algorithm: an independent sample $X_N = \{x_1, \ldots, x_N\}$ from the uniform distribution on $\mathfrak{X}$ is generated and a local optimization algorithm $\mathcal{D}$ is applied at each $x_j \in X_N$. Let N_i $(i = 1, \ldots, l)$ be the number of points $x_j \in X_N$ belonging to A_i; that is, N_i is the number of descents to $x_*^{(i)}$ from the points $x_1, \ldots, x_N$. According to the definition, $N_i \geqslant 0$ $(i = 1, \ldots, l)$, $\sum_{i=1}^{l} N_i = N$, and the random vector $(N_1, \ldots, N_l)$ follows the multinomial distribution

$$\Pr\{N_1 = n_1, \ldots, N_l = n_l\} = \binom{N}{n_1, \ldots, n_l} \theta_1^{n_1} \ldots \theta_l^{n_l},$$

where

$$\sum_{i=1}^{l} n_i = N, \quad \binom{N}{n_1, \ldots, n_l} = \frac{N!}{n_1! \ldots n_l!}, \quad n_i \geqslant 0 \quad (i = 1, \ldots, l).$$

If l is known, then the problem of drawing statistical inferences concerning the number of local minimizers l, the parameter vector $\theta = (\theta_1, \ldots, \theta_l)$, and the number N_* of trials that guarantees with a given probability that all local minimizers are found, is the standard problem of making statistical inferences about parameters of a multinomial distribution. This problem is well documented in literature, see e.g. Chap. 35 in [2]. The main difficulty is caused by the fact that l is usually unknown. If an upper bound for l is known, then one can apply standard statistical methods; if an upper bound for l is unknown, the Bayesian approach is a natural alternative. Both of these methods are comprehensively studied and explained in [13]; see also [12, Section 2.6.2].

In relation to the algorithm METOD described and discussed in Section 3, we are more interested in whether we have succeeded in finding the global minimizer taking into account that some of our decisions about early termination of local descents may be erroneous. Assume that $\mathrm{vol}(\mathfrak{X}) = 1$ and $\mathrm{vol}(A^*) = \alpha > 0$, where A^* is the region of attraction of the global minimizer (the value of α does not have to be known, of course). In view of the difficulties discussed in Section 3.3, local descents, including the ones which would lead to the global minimizer, can be stopped early and the corresponding initial points assigned to a wrong region of attraction. As noted in Section 2.2 the probability of this event is related to the number of checks of the inequality (4) and the degree of local non-convexity of the objective function. As follows from the discussion at the end of Section 2.2, the probability of the fact that an initial point is assigned to a wrong region of attraction is very roughly $1/2^k$, where k is the number of checks of the inequalities (9).

So we end up with the following rather simple situation. We have a Bernoulli trial with success probability α (when our uniformly distributed starting point x_n belongs to A^*) but on top of this we have a drop-out event (happening with rather small probability κ, which we for simplicity assume the same for all decisions) where we will reassign this initial point to another region of attraction. Therefore, each starting point x_n (taken randomly and independently of the other points) will be assigned to A^* with probability at least $\delta = \alpha(1-\kappa)$. The first starting point assigned to A^* will create a full local descent trajectory converging to the global minimizer. Note that after finding the first point in A^*, the probability of assigning starting points to A^* will increase from $\delta = \alpha(1-\kappa)$ to $\delta' = \alpha(1-\kappa) + (1-\alpha)\kappa/(L-1)$, where L is the number of local minimizers found so far, as there appears a new possibility of assigning points to A^* when they do not belong there. We can ignore this as we are only interested in the event that at least one initial point will be assigned to A^* and hence that the global minimizer is found.

With N initial i.i.d. uniform starting points, the probability of finding the global minimum is $p_{\delta,N} = 1 - (1-\delta)^N$. Let $N_{\delta,\gamma}$ be the smallest N such that $p_{\delta,N} \geq 1 - \gamma$; that is, if we choose $N \geq N_{\delta,\gamma}$ then we would guarantee that the probability of finding the global minimizer is at least $1 - \gamma$. Solving the inequality $p_{\delta,N} \geq 1 - \gamma$ with respect to N we find

$$N_{\delta,\gamma} = \left\lceil \frac{\log \gamma}{\log(1-\delta)} \right\rceil . \tag{10}$$

Table 4 shows some values of $N_{\delta,\gamma}$. From this table we can conclude that there is very little hope of finding the global minimum if the volume of attraction of the global minimizer is smaller than 0.00001. On the other hand, if $\delta \leq 0.001$ then METOD would not require many starting points for guaranteeing high probabilities of finding the global minimizer.

	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$	$r=7$	$r=8$
$\gamma=0.05$	29	299	2995	29956	299572	2995731	29957322	299573226
$\gamma=0.01$	44	459	4603	46050	460515	4605168	46051700	460517017

Table 1 Values of $N_{\delta,\gamma}$, see (10), for $\gamma=0.01$ and 0.05, $\delta=10^{-r}$, $r=1,2,\ldots,8$.

5 Numerical results

5.1 Objective function 1: Minimum of several quadratic forms

In numerical experiments, we have considered two families of the objective function. The first family of objective functions is ideally suited for our algorithm as the objective function is always exactly quadratic in regions of attraction of all local minimizers:

$$f(x)=\min_{1\le p\le P}(x-x_{0p})^T A_p^T \Sigma_p A_p (x-x_{0p}), \tag{11}$$

where P is the number of minima; A_p $(p=1,...,P)$ are randomly chosen rotation matrices of size $d\times d$; Σ_p $(p=1,...,P)$ are diagonal positive definite matrices of size $d\times d$; x_{0p} $(p=1,...,P)$ are random points in $\mathfrak{X}$. All minima of (11) are (equally) global, and for this function it is desirable to find all minimizers.

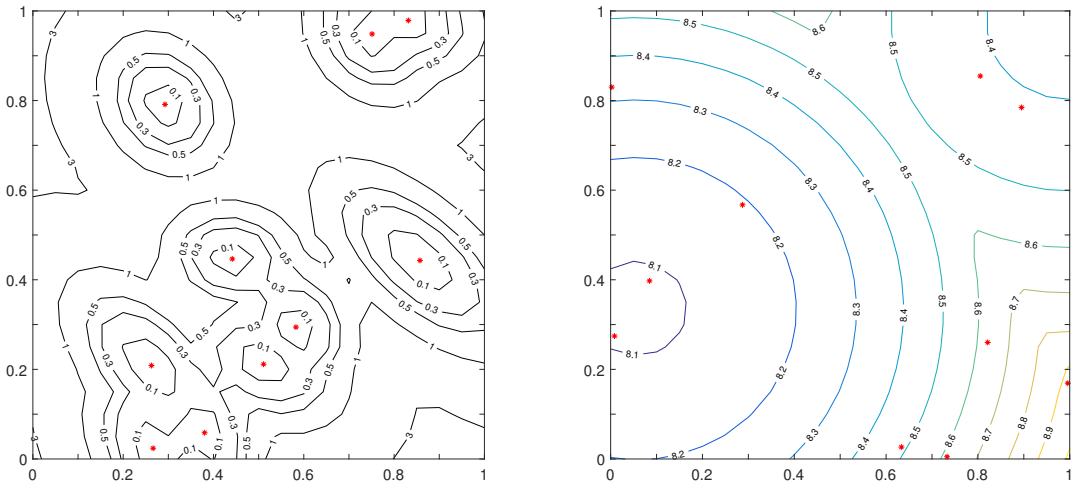

Fig. 1 The contour lines of a function of the form (11) with $d=2$ (left) and the contour lines of a two-dimensional section of a function of the form (11) with $d=100$ (right). Stars denote the minimizers in the case $d=2$ and the projections of the minimizers to the secant plane in the case $d=100$.

The contours of a two-dimensional function (11), and of a two-dimensional section of a hundred-dimensional function (11) are presented in Figure 1 in the cases when the matrices $A_p^T \Sigma_p A_p$ are the same (different in two dimensions, of course); the contours show that the surfaces are quite different despite in both cases $P = 10$. Typical steepest descent iterations initialized at random starting points are displayed in Fig. 2 for the case $d = 2$ and function (11) with $P = 6$.

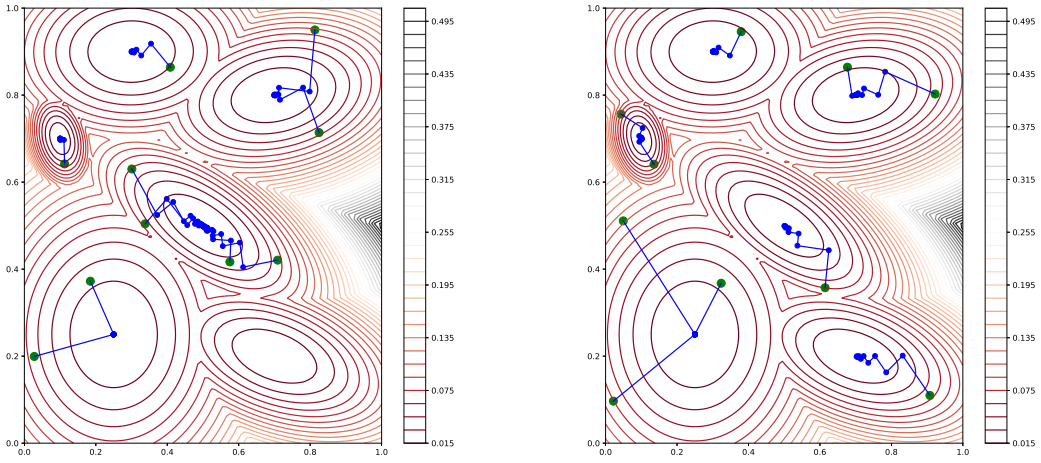

Fig. 2 Typical steepest decent iterations for the case $d = 2$ with function (11) initialized at two different sets of ten starting points for the objective function (11). The minimizers are $(0.1, 0.7)$, $(0.3, 0.9)$, $(0.7, 0.8)$, $(0.7, 0.2)$, $(0.5, 0.5)$ and $(0.25, 0.25)$.

Standard multistart and METOD were compared with respect to the number of evaluations of the objective function values needed to solve a problem, i.e. to find all minima. The cases $d = 2$ and $d = 100$ were considered. The same 1000 randomly generated test functions (11) were minimized by multistart and METOD. The average condition number of the test functions was equal to 2.11 (standard deviation 0.31) in the case $d = 2$, and equal to 3.23 (standard deviation was equal to 0.82) in the case $d = 100$. The starting points for the local descent for both algorithms were the same. The termination condition was defined either by the minimum norm of gradient equal to 10^{-7}, or by the minimum step length equal to 10^{-5}; note that in the majority of cases the termination of METOD was due to the latter condition.

The average number of function evaluations ($\bar{nf}$) and the average number of local descents ($\bar{nd}$) are presented in Table 2 as well as the corresponding standard deviations σ_{nf} and σ_{nd}. In the case $d = 2$, the average number of local descents needed for multistart was smaller about 2/3 of that needed for METOD. Some local descents, despite moving to an undiscovered minimizer, were terminated because they closely approached an earlier found minimizer. Therefore, METOD continued after multistart completed the global minimization. Multistart found nine minima out of ten in three cases, and METOD in four cases.

The advantage of METOD was much more impressive in the case $d = 100$ where the total number (for all 1000 functions) of started local searches was 48291 and 48298 for the multistart and METOD respectively. The average number of objective function evaluations was equal to 1581 for multistart

	Multistart				METOD			
d	nf	σ_{nf}	nd	σ_{nd}	nf	σ_{nf}	nd	σ_{nd}
2	605.3	300.3	39.3	19.7	404.3	166.1	40.4	20.5
100	1581.0	900.8	48.3	27.6	433.9	73.5	48.3	2.6

Table 2 The computational resources needed to the multistart and METOD to minimize the objective functions (11).

and 433.9 for METOD. Other statistical data are presented in Table 2. Nine minima (out of ten) were found by the multistart 4 times, and by METOD 5 times.

The histograms in Figures 3 and 4 complement Table 2 where the presented parameters do not fully represent the corresponding distributions in view of the non-symmetry of the histograms.

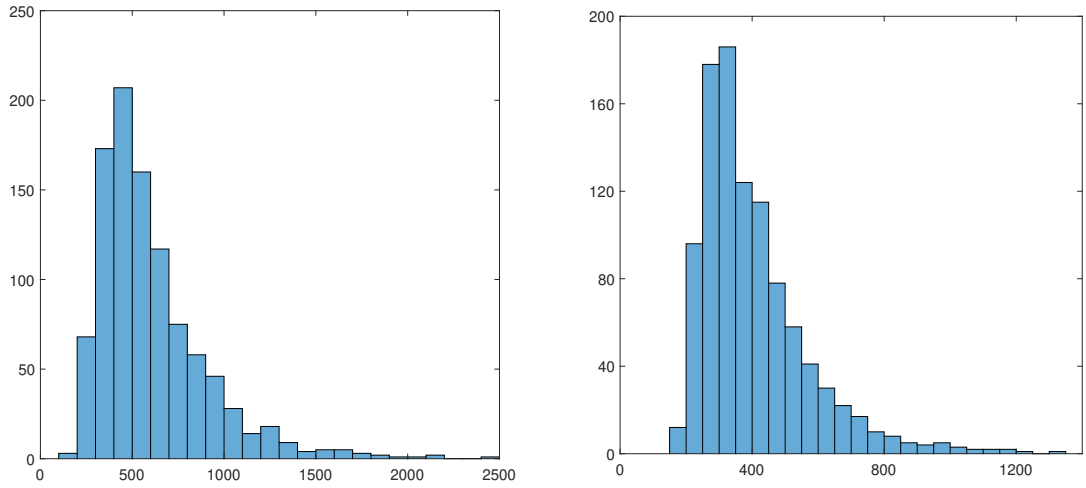

Fig. 3 The histograms of the number of computations of the objective function ($d = 2$) values: by the multistart (left) and by METOD (right).

Different values of the algorithm parameters, M and β, are tested for the objective functions (11), to see the effect they have on correctly classifying each starting point to the correct region of attraction. A single experiment consists of choosing random function parameters for the objective functions (11), and a set of random starting points in $\mathfrak{X}$. The algorithm then classifies each starting point to a region of attraction, A_l. To check if the algorithm has correctly classified a starting point, the local minima is found by using iterations (7) and stopped when $\|\nabla f(x_n^{(k)})\| < \delta$. The region of attraction the local minima belongs to is then compared to the region of attraction assigned by the algorithm.

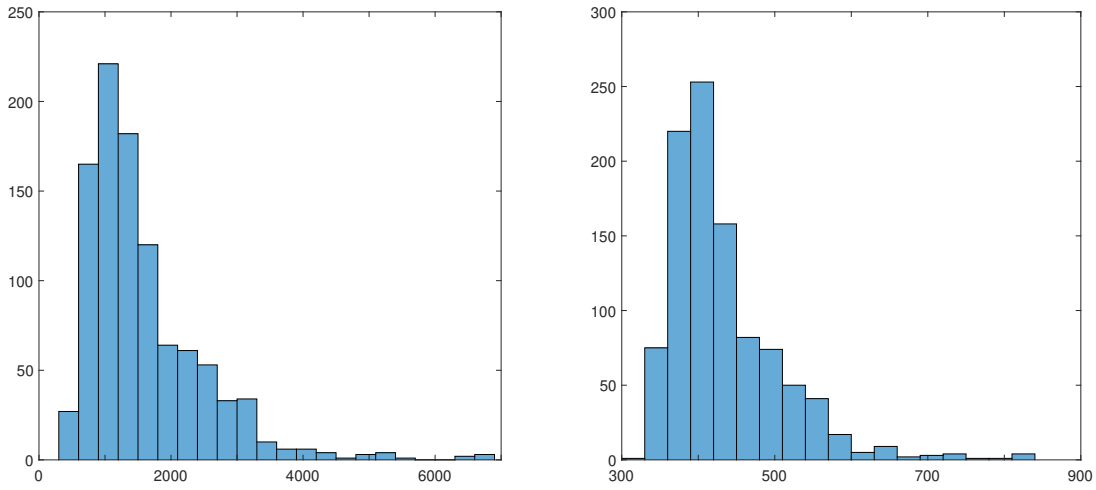

Fig. 4 The histograms of the number of objective function evaluations for $d = 100$: by the multistart (left) and by METOD (right).

For the class of objective functions (11), Tables 3 and 4 display the total number of regions of attraction identified, the number of times a point did not satisfy the inequalities (9) and consequently the number of descents (7) that were applied to find the corresponding local minima and the total percentage of misclassifications of points for dimensions $d = 50$ and $d = 100$.

β	M	Number of regions of attraction			Number of descents			Total % of misclassifications
		50	49	≤ 48	50	49	≤ 48	
0.005	2	27	33	40	27	33	40	0.084
	3	31	38	31	31	38	31	0.028
0.01	2	29	31	40	29	31	40	0.076
	3	33	37	30	33	37	30	0.023
0.05	2	36	34	30	36	34	30	0.025
	3	39	34	27	39	34	27	0.005
0.1	2	41	31	28	41	31	28	0.005
	3	41	33	26	41	33	26	0.001

Table 3 Outputs for 100 random functions from (11) with $d = 50$, $N = 1000$ and $P = 50$.

Tables 3 and 4 show that the total percentage of misclassifications decreases as d increases. Also, each time a point did not satisfy the inequalities (9) and consequently iterations (7) were applied to find the corresponding local minima, this always resulted in a new region of attraction being

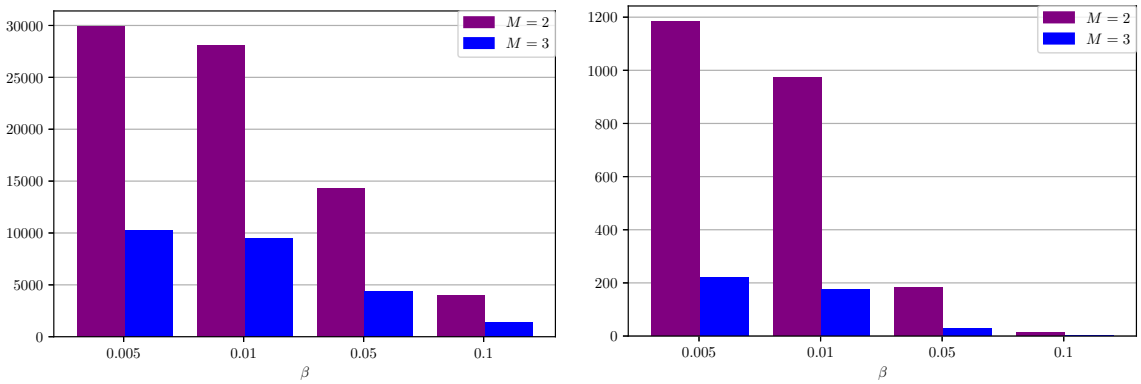

Fig. 5 Total number of points satisfying step (4b) of the METOD Algorithm for 100 random functions from (11) for $d = 50$ (left) and $d = 100$ (right), with $N = 1000$ and $P = 50$

β	M	Number of regions of attraction			Number of descents			Total % of misclassifications
		50	49	≤ 48	50	49	≤ 48	
0.005	2	34	31	35	34	31	35	0.001
	3	34	32	34	34	32	34	0
0.01	2	34	31	35	34	31	35	0.001
	3	34	32	34	34	32	34	0
0.05	2	34	32	34	34	32	34	0
	3	34	32	34	34	32	34	0
0.1	2	34	32	34	34	32	34	0
	3	34	32	34	34	32	34	0

Table 4 Outputs for 100 random functions from (11) with $d = 100$, $N = 1000$ and $P = 50$.

identified. Figure 5 shows the total number of points satisfying step (4b) of the algorithm decreases as M and β increase. When $d = 50$, the total number of points satisfying step (4b) of the algorithm is much higher in comparison to when $d = 100$. It may be of consideration to alter step (4b) of the algorithm to apply an iteration of local descent to x_n if the set S_n contains more than one index l and repeating this until set S_n has exactly one index l or is empty. However, from the results shown in Table 4, the alteration of step (4b) would not improve results, as the single misclassification occurs from (9) being satisfied for a single region of attraction which the point did not belong to.

5.2 Objective function 2: Weighted sum of Gaussian densities

The second family is a constant minus a weighted sum of Gaussian densities:

$$f(x) = C - \sum_{p=1}^{P} c_p \exp\left\{-\frac{1}{2\sigma^2}(x - x_{0p})^T A_p^T \Sigma_p A_p (x - x_{0p})\right\}. \tag{12}$$

Here we have: C is a constant (served only for making plots looking more attractive); P is the number of Gaussian densities; A_p $(p = 1, ..., P)$ are randomly chosen rotation matrices of size $d \times d$; Σ_p $(p = 1, ..., P)$ are diagonal positive definite matrices of size $d \times d$; x_{0p} $(p = 1, ..., P)$ are random points in $\mathfrak{X}$ (centers of the Gaussian densities); c_p $(p = 1, ..., P)$ are fixed constants.

The function (12) has the gradient

$$\nabla f(x) = \sum_{p=1}^{P} \frac{c_p}{\sigma^2} \exp\left\{-\frac{1}{2\sigma^2}(x - x_{0p})^T A_p^T \Sigma_p A (x - x_{0p})\right\} A_p^T \Sigma_p A (x_n - x_{0p}) \,.$$

Typical steepest descent iterations initialized at random starting points are displayed in Fig. 6 for the case $d = 2$ and function (12) with $P = 6$, $\sigma^2 = 1/50$, $C = 0$.

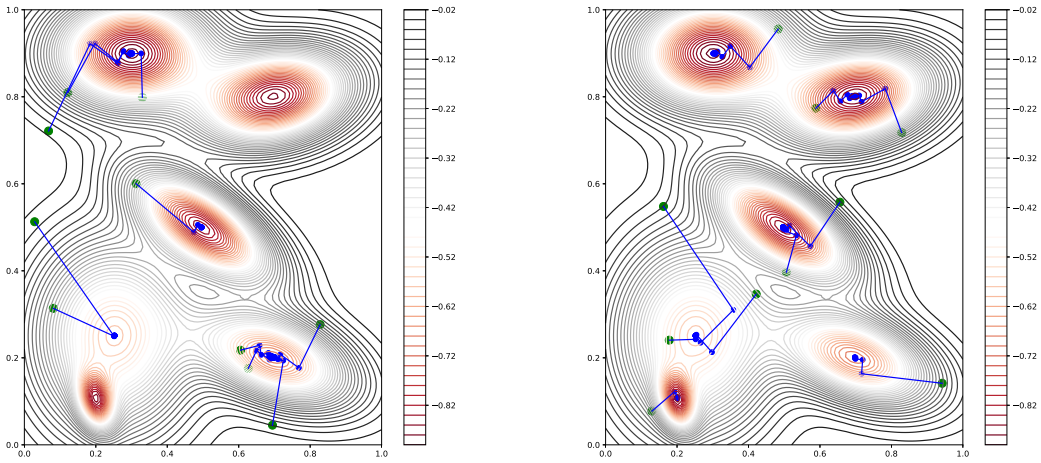

Fig. 6 Typical steepest decent iterations for the case $d = 2$ initialized at two different sets of ten starting points for objective function (12). Local minima at $(0.2, 0.1)$, $(0.3, 0.9)$, $(0.7, 0.8)$, $(0.7, 0.2)$, $(0.5, 0.5)$ and $(0.25, 0.25)$.

For the class of objective functions (12), Tables 5 and 6 show the total number of regions of attraction identified; the average, minimum and maximum number of times a point did not satisfy the inequalities (9) and consequently the number of descents (7) that were applied to find the corresponding local minima and the total percentage of misclassifications of points for dimensions $d = 50$ and $d = 100$.

Tables 5 and 6 show that the total percentage of misclassifications decreases as d increases. Also, the smaller the values of β and M, the higher the total percentage of misclassifications. Although there are some misclassifications, no misclassifications result in a region of attraction being missed. In some cases less than 20 regions of attraction are found and this is due to the insufficient number of starting points.

It can be observed from Figure 7 that the total number of points satisfying step (4b) of the algorithm is much lower in comparison to Figure 5. Even though some points still do satisfy (4b) of the algorithm, this does not result in any regions of attraction being missed and is due to the insufficient number of starting points. For Table 6, only one point out of 100,000 satisfied step (4b) of the algorithm. This occurred for the algorithm parameters $M = 4$ and $\beta = 0.005$ or $\beta = 0.01$, but it did not result in a misclassification. Hence, altering the condition (4b) would not improve results observed in Tables 5 and 6.

β	M	Number of regions			Descents			Total % of misclassifications
		20	19	≤ 18	Min	Max	Avg	
0.005	2	95.0	5.0	0.0	355.0	693.0	500.01	0.135
	3	95.0	5.0	0.0	48.0	337.0	105.98	0.032
	4	95.0	5.0	0.0	21.0	84.0	29.09	0.007
0.01	2	95.0	5.0	0.0	357.0	694.0	501.48	0.131
	3	95.0	5.0	0.0	48.0	337.0	106.34	0.032
	4	95.0	5.0	0.0	21.0	84.0	29.13	0.007
0.05	2	95.0	5.0	0.0	364.0	707.0	514.48	0.110
	3	95.0	5.0	0.0	53.0	353.0	108.75	0.029
	4	95.0	5.0	0.0	21.0	85.0	29.42	0.004
0.1	2	95.0	5.0	0.0	380.0	723.0	530.58	0.100
	3	95.0	5.0	0.0	53.0	371.0	111.92	0.023
	4	95.0	5.0	0.0	21.0	86.0	29.73	0.004

Table 5 Outputs for 100 random functions from (12) with $d = 50$, $N = 1000$, $P = 20$. and $\sigma^2 = \frac{4}{3}$.

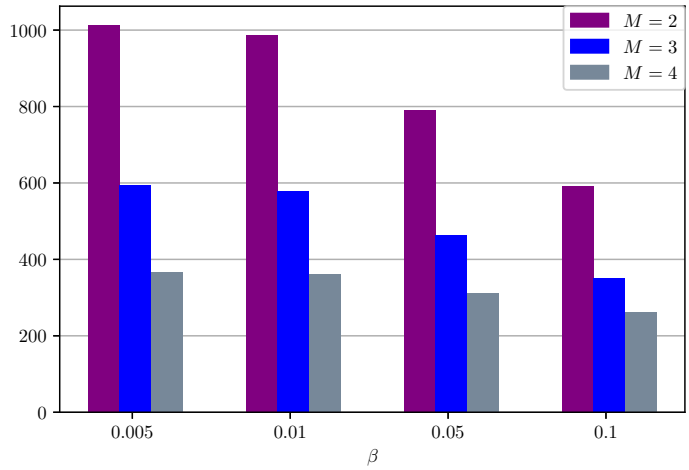

Fig. 7 Total number of points satisfying step (4b) of the METOD Algorithm for 100 random functions from (12) with $d = 50$, $N = 1000$, $P = 20$ and $\sigma^2 = \frac{4}{3}$

Acknowledgements

The work of A.Zilinskas was supported by the Research Council of Lithuania under Grant No. P-MIP-17-61. The work of A.Zhigljavsky was supported by a grant of Crimtan Holding Limited.

References

1. R. Haycroft, L. Pronzato, H.P. Wynn, and A. Zhigljavsky. Studying convergence of gradient algorithms via optimal experimental design theory. In *Optimal design and related areas in optimization and statistics*, pages 13–37. Springer, 2009.
2. N.L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions.* Wiley, New York, 1997.

β	M	Number of regions			Descents			Total % of misclassifications
		20	19	≤ 18	Min	Max	Avg	
0.005	2	60.0	28.0	12.0	686.0	973.0	888.15	0.001
	3	60.0	28.0	12.0	296.0	807.0	560.78	0.001
	4	60.0	28.0	12.0	93.0	558.0	239.34	0.000
0.01	2	60.0	28.0	12.0	688.0	974.0	888.62	0.001
	3	60.0	28.0	12.0	296.0	807.0	561.12	0.001
	4	60.0	28.0	12.0	93.0	558.0	239.44	0.000
0.05	2	60.0	28.0	12.0	691.0	976.0	892.19	0.001
	3	60.0	28.0	12.0	298.0	807.0	563.69	0.001
	4	60.0	28.0	12.0	93.0	558.0	240.87	0.000
0.1	2	60.0	28.0	12.0	693.0	978.0	896.46	0.001
	3	60.0	28.0	12.0	299.0	809.0	567.19	0.001
	4	60.0	28.0	12.0	93.0	558.0	242.37	0.000

Table 6 Outputs for 100 random functions from (12) with $d = 100$, $N = 1000$, $P = 20.$ and $\sigma^2 = 4$.

3. T. Krityakierne and C. A. Shoemaker. Soms: Surrogate multistart algorithm for use with nonlinear programming for global optimization. *International Transactions in Operational Research*, 24(5):1139–1172, 2017.
4. D. López-Soto, F. Angel-Bello, S. Yacout, and A. Alvarez. A multi-start algorithm to design a multi-class classifier for a multi-criteria abc inventory classification problem. *Expert Systems with Applications*, 81:12–21, 2017.
5. M. Lozano, F. J. Rodriguez, D. Peralta, and C. García-Martínez. Randomized greedy multi-start algorithm for the minimum common integer partition problem. *Engineering Applications of Artificial Intelligence*, 50:226–235, 2016.
6. D. Peri and F. Tinti. A multistart gradient-based algorithm with surrogate model for global optimization. *Communications in Applied and Industrial Mathematics*, 3(1), 2012.
7. L. Pronzato, H. P. Wynn, and A. A. Zhigljavsky. *Dynamical Search: Applications of Dynamical Systems in Search and Optimization*. CRC Press, 1999.
8. S. Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747*, 2016.
9. D. Taubman and A. Zakhor. A multi-start algorithm for signal adaptive subband systems (image coding). In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 213–216. IEEE, 1992.
10. A. Törn and A. Žilinskas. *Global Optimization*. Springer, 1989.
11. W. Tu and W. Mayne. Studies of multi-start clustering for global optimization. *Int. J. Numer. Meth. Engng*, 53:2239 — 2252, 2002.
12. A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*. Springer, 2008.
13. R. Zieliński. A statistical estimate of the structure of multi-extremal problems. *Mathematical Programming*, 21(1):348–356, 1981.